Article

AI models collapse when trained on recursively generated data

https://doi.org/10.1038/s41586-024-07566-y

Received: 20 October 2023

Accepted: 14 May 2024

Published online: 24 July 2024

Open access

Check for updates

Ilia Shumailov[1,8]✉, Zakhar Shumaylov[2,8]✉, Yiren Zhao[3], Nicolas Papernot[4,5], Ross Anderson[6,7,9] & Yarin Gal[1]✉

Stable diffusion revolutionized image creation from descriptive text. GPT-2 (ref. 1), GPT-3(.5) (ref. 2) and GPT-4 (ref. 3) demonstrated high performance across a variety of language tasks. ChatGPT introduced such language models to the public. It is now clear that generative artificial intelligence (AI) such as large language models (LLMs) is here to stay and will substantially change the ecosystem of online text and images. Here we consider what may happen to GPT-{*n*} once LLMs contribute much of the text found online. We find that indiscriminate use of model-generated content in training causes irreversible defects in the resulting models, in which tails of the original content distribution disappear. We refer to this effect as 'model collapse' and show that it can occur in LLMs as well as in variational autoencoders (VAEs) and Gaussian mixture models (GMMs). We build theoretical intuition behind the phenomenon and portray its ubiquity among all learned generative models. We demonstrate that it must be taken seriously if we are to sustain the benefits of training from large-scale data scraped from the web. Indeed, the value of data collected about genuine human interactions with systems will be increasingly valuable in the presence of LLM-generated content in data crawled from the Internet.

The development of LLMs is very involved and requires large quantities of training data. Yet, although current LLMs[2,4–6], including GPT-3, were trained on predominantly human-generated text, this may change. If the training data of most future models are also scraped from the web, then they will inevitably train on data produced by their predecessors. In this paper, we investigate what happens when text produced by, for example, a version of GPT forms most of the training dataset of following models. What happens to GPT generations GPT-{*n*} as *n* increases? We discover that indiscriminately learning from data produced by other models causes 'model collapse'—a degenerative process whereby, over time, models forget the true underlying data distribution, even in the absence of a shift in the distribution over time. We give examples of model collapse for GMMs, VAEs and LLMs. We show that, over time, models start losing information about the true distribution, which first starts with tails disappearing, and learned behaviours converge over the generations to a point estimate with very small variance. Furthermore, we show that this process is inevitable, even for cases with almost ideal conditions for long-term learning, that is, no function estimation error. We also briefly mention two close concepts to model collapse from the existing literature: catastrophic forgetting arising in the framework of task-free continual learning[7] and data poisoning[8,9] maliciously leading to unintended behaviour. Neither is able to explain the phenomenon of model collapse fully, as the setting is fundamentally different, but they provide another perspective on the observed phenomenon and are discussed in more depth in the Supplementary Materials. Finally, we discuss the broader implications of model collapse. We note that access to the original data distribution is crucial: in learning tasks in which the tails of the underlying distribution matter, one needs access to real human-produced data. In other words, the use of LLMs at scale to publish content on the Internet will pollute the collection of data to train their successors: data about human interactions with LLMs will be increasingly valuable.

What is model collapse?

Definition 2.1 (model collapse). Model collapse is a degenerative process affecting generations of learned generative models, in which the data they generate end up polluting the training set of the next generation. Being trained on polluted data, they then mis-perceive reality. The process is depicted in Fig. 1a. We separate two special cases: early model collapse and late model collapse. In early model collapse, the model begins losing information about the tails of the distribution; in late model collapse, the model converges to a distribution that carries little resemblance to the original one, often with substantially reduced variance.

This process occurs owing to three specific sources of error compounding over generations and causing deviation from the original model:

- **Statistical approximation error.** This is the primary type of error, which arises owing to the number of samples being finite, and disappears as the number of samples tends to infinity. This occurs because

[1]OATML, Department of Computer Science, University of Oxford, Oxford, UK. [2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK. [3]Department of Electrical and Electronic Engineering, Imperial College London, London, UK. [4]University of Toronto, Toronto, Ontario, Canada. [5]Vector Institute, Toronto, Ontario, Canada. [6]Department of Computer Science and Technology, University of Cambridge, Cambridge, UK. [7]School of Informatics, University of Edinburgh, Edinburgh, UK. [8]These authors contributed equally: Ilia Shumailov, Zakhar Shumaylov. [9]Deceased: Ross Anderson. ✉e-mail: ilia.shumailov@chch.ox.ac.uk; zs334@cam.ac.uk; yarin@cs.ox.ac.uk

Content courtesy of Springer Nature, terms of use apply. Rights reserved

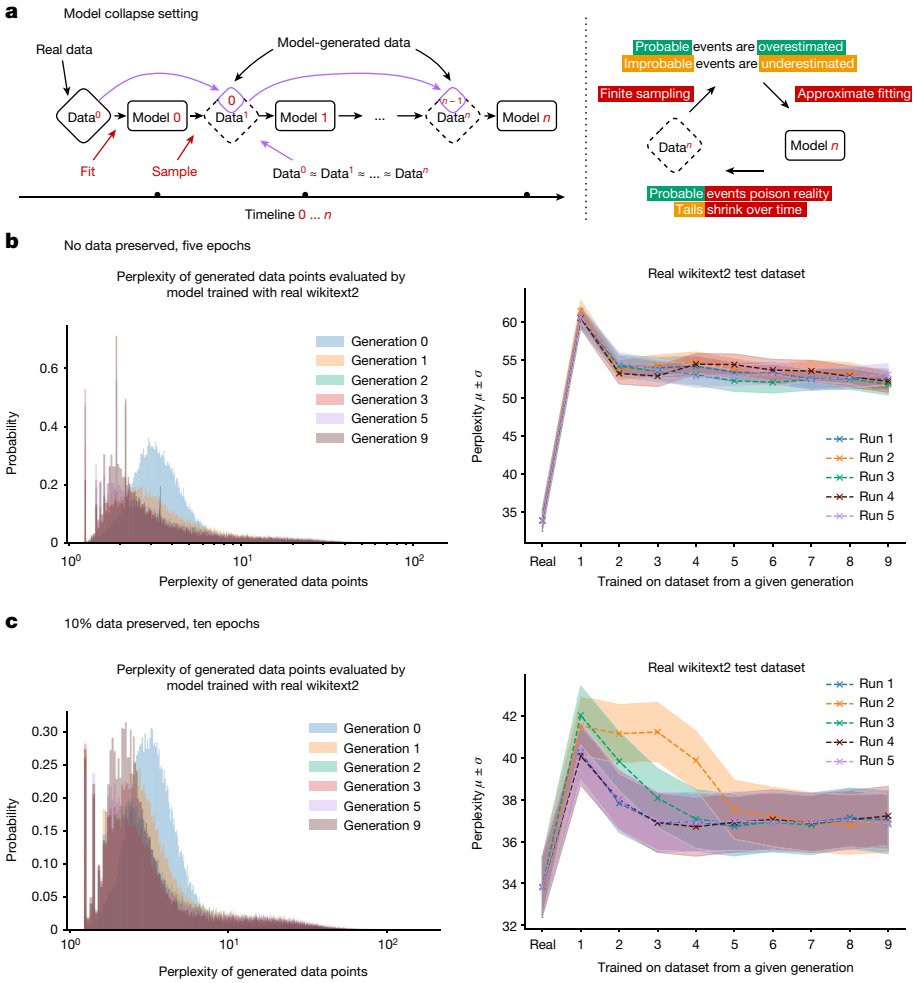

Fig. 1 | The high-level description of the feedback mechanism in the learning process. a, Model collapse refers to a degenerative learning process in which models start forgetting improbable events over time, as the model becomes poisoned with its own projection of reality. Here data are assumed to be human-curated and start off clean; then model 0 is trained and data are sampled from it; at step n, data are added to the overall data from step $n-1$ and this combination is used to train model n. Data obtained with Monte Carlo sampling should ideally be statistically close to the original, provided that fitting and sampling procedures are perfect. This process depicts what happens in real life with the Internet: model-generated data become pervasive. **b**,**c**, Performance of OPT-125m models of different generations evaluated using the original wikitext2 test dataset. Shown on the left are the histograms of perplexities of each individual data training sequence produced by different generations as evaluated by the very first model trained with the real data. Over the generations, models tend to produce samples that the original model trained with real data is more likely to produce. At the same time, a much longer tail appears for later generations. Later generations start producing samples that would never be produced by the original model, that is, they start misperceiving reality based on errors introduced by their ancestors. The same plots are shown in 3D in the Supplementary Materials. On the right, average perplexity and its standard deviation are shown for each independent run. The x axis refers to the generation of the model. 'Real' refers to the 'model 0' trained on the original wikitext2 dataset; model 1 was trained on the data produced by model 0, model 2 was trained on data produced by model 1 and so on, with all generated datasets equal in size. We find that models trained on generated data are able to learn some of the original task, but with errors, as seen from the increase in perplexity.

of a non-zero probability that information can get lost at every step of resampling.

- **Functional expressivity error.** This is a secondary type of error, arising owing to limited function approximator expressiveness. In particular, neural networks are only universal approximators as their size goes to infinity. As a result, a neural network can introduce non-zero likelihood outside the support of the original distribution or zero likelihood inside the support of the original distribution. A simple example of the expressivity error is if we tried fitting a mixture of two Gaussians with a single Gaussian. Even if we have perfect information about the data distribution (that is, infinite number of samples), model errors will be inevitable. However, in the absence of the other two types of error, this can only occur at the first generation.
- **Functional approximation error.** This is a secondary type of error, arising primarily from the limitations of learning procedures, for example, structural bias of stochastic gradient descent[10,11] or choice of objective[12]. This error can be viewed as one arising in the limit of infinite data and perfect expressivity at each generation.

Each of the above can cause model collapse to get worse or better. More approximation power can even be a double-edged sword—better expressiveness may counteract statistical noise, resulting in a good approximation of the true distribution, but it can equally compound the noise. More often than not, we get a cascading effect, in which individual inaccuracies combine to cause the overall error to grow. For example, overfitting the density model causes the model to extrapolate incorrectly and assigns high-density regions to low-density regions not covered in the training set support; these will then be sampled with arbitrary frequency. It is worth noting that other types of error exist. For example, computers have limited precision in practice. We now turn to mathematical intuition to explain how the above give rise

Content courtesy of Springer Nature, terms of use apply. Rights reserved

to the errors observed, how different sources can compound and how we can quantify the average model divergence.

Theoretical intuition

Here we provide a theoretical intuition for the phenomenon of model collapse. We argue that the process of model collapse is universal among generative models that recursively train on data generated by previous generations. We quantify the sources of errors discussed in the previous section by examining two mathematical models, which prove to be simple enough to provide analytical expressions for quantities of interest, but also portray the phenomenon of model collapse: a discrete distribution in the absence of functional expressivity and approximation errors, and a multidimensional Gaussian approximation, portraying joint functional expressivity and statistical errors. We further illustrate the impact of all three jointly for a more complex setting of density estimation in Hilbert spaces in the Supplementary Materials.

The overall stochastic process we consider, which we call learning with generational data, is the following. The dataset at generation i is $\mathcal{D}_i$, comprising independent and identically distributed random variables X^i_j with distribution $p_i, j \in \{1,\ldots,M_i\}$ denotes the size of the dataset. Going from generation i to generation $i+1$, we aim to estimate the distribution of samples in $\mathcal{D}_i$, with an approximation $p_{\theta_{i+1}}$. This step is what we refer to as functional approximation, $p_{\theta_{i+1}} = \mathcal{F}_\theta(p_i)$. The dataset $\mathcal{D}_{i+1}$ is then generated by sampling from $p_{i+1} = \alpha_i p_{\theta_{i+1}} + \beta_i p_i + \gamma_i p_0$, with non-negative parameters $\alpha_i, \beta_i, \gamma_i$ summing to 1, that is, they represent proportions of data used from different generations. This corresponds to a mixing of data coming from the original distribution (γ_i), data used by the previous generation (β_i) and data generated by the new model (α_i). We refer to this as the sampling step. For the mathematical models to come, we consider $\beta_i = \gamma_i = 0$, that is, data only from a single step are used, whereas numerical experiments are performed on more realistic choices of parameters.

Discrete distributions with exact approximation

In this subsection, we consider a discrete probability distribution in absence of functional approximation and expressivity errors, that is, $\mathcal{F}(p) = p$. In this case, model collapse arises only because of statistical errors from the sampling step. At first, the tails (low-probability events) begin to disappear as a result of the low probability of sampling them and, over time, support of the distribution shrinks. Denoting the sample size as M, if we consider state i with probability $q \leq \frac{1}{M}$, the expected number of samples with value i coming from those events will be less than 1. In practice, this would mean that we lose information about them. Considering more generally some state i with probability q, using standard conditional probability, we can show that the probability of losing information (that is, sampling no data at some generation) is equal to $1-q$, implying that the distribution must converge to a delta function positioned at some state, with the probability of ending up at a certain state equal to the probability of sampling said state from the original distribution.

This can be shown directly by considering the process $\mathbf{X}^i \to \mathcal{F} \to p_{i+1} \to \mathbf{X}^{i+1}$ as a Markov chain, as $\mathbf{X}^{i+1}$ only depends on $\mathbf{X}^i$. Furthermore, if all the X^i_j have the same value, then at the next generation, the approximated distribution will be exactly a delta function and therefore all of X^{i+1}_j will also have the same value. This implies that the Markov chain contains at least one absorbing state and therefore, with probability 1, it will converge to one of the absorbing states. This is a well-known fact, of which a proof is provided in the Supplementary Materials. For this chain, the only absorbing states are those corresponding to delta functions. As a result, as we follow the progress of model collapse, we are guaranteed to end up in a constant state, having lost all the information of the original distribution when the chain is absorbed. This argument also works in general owing to floating-point representations being discrete, making the Markov chain over the parameters of the model discrete. Thus, as long as the model parameterization allows for delta functions, we will get to it, because—owing to sampling errors—the only possible absorbing states are delta functions. On the basis of the discussion above, we see how both early model collapse, in which only the low-probability events get cut off, and late stage model collapse, in which the process begins to collapse into a single mode, must arise in the case of discrete distributions with perfect functional approximation.

Multidimensional Gaussian

Following the discussion about discrete distributions, we now present a more generic result, which can be shown in the Gaussian approximation setting, in which each generation is approximated using the unbiased estimates of the mean and the variance. A similar result holds more generally, which we detail in the Supplementary Materials.

Theorem 3.1 (Gaussian model collapse). Assume the original data are sampled from distribution $\mathcal{D}_0$ (not necessarily Gaussian), with non-zero sample variance. Assume X^n are fit recursively using the unbiased sample mean and variance estimators from the previous generation, $X^n_j | \mu_n, \Sigma_n \sim \mathcal{N}(\mu_n, \Sigma_n)$, with a fixed sample size. Then,

$$\mathbb{E}[\mathbb{W}_2^2(\mathcal{N}(\mu_n, \Sigma_n), \mathcal{D}_0)] \to \infty; \quad \Sigma_n \overset{\text{a.s.}}{\to} 0 \text{ as } n \to \infty,$$

in which $\mathbb{W}_2$ denotes the Wasserstein-2 distance between the true distribution and its approximation at generation n.

In words, this implies that not only does the nth generation approximation diverge arbitrarily far from the original one but it also collapses to be zero variance as the number of generations increases, with probability 1. The results are very analogous to that seen in the discrete case, with this theorem illustrating the effect of late stage model collapse, in which the process begins to collapse to be zero variance. The early stage model collapse can also be seen and the interested reader is referred to the Supplementary Materials for a more in-depth discussion.

Model collapse in language models

In this section, we evaluate the effect of model collapse on language models. We cover more interpretable machine learning models—VAEs and GMMs—in the Supplementary Materials. Code is publically available in ref. 13.

Model collapse is universal across various families of machine learning models. Yet, if small models such as GMMs and VAEs are normally trained from scratch, LLMs are different. They are so expensive to retrain from scratch that they are typically initialized with pre-trained models such as BERT[4], RoBERTa[5] or GPT-2 (ref. 2), which are trained on large text corpora. They are then fine-tuned to various downstream tasks[14].

Here we explore what happens with language models when they are sequentially fine-tuned with data generated by other models. We can easily replicate all experiments covered in this paper with larger language models in non-fine-tuning settings to demonstrate model collapse. Given that training a single moderately large model produces twice the American lifetime's worth of CO_2 (ref. 15), we opted to not run such an experiment and instead focus on a more realistic setting for a proof of concept. Note that even the language experiments described in this paper took weeks to run. We evaluate the most common setting of training a language model—a fine-tuning setting for which each of the training cycles starts from a pre-trained model with recent data. The data here come from another fine-tuned pre-trained model. Because training is restricted to produce models that are close to the original pre-trained model, and data points generated by the models will generally produce very small gradients, the expectation here may be that the model should only change moderately after fine-tuning. We fine-tune the OPT-125m causal language model made available by Meta through Hugging Face[6].

Content courtesy of Springer Nature, terms of use apply. Rights reserved

We fine-tune it on the wikitext2 dataset[16]. For data generation from the trained models, we use a five-way beam search. We block training sequences to be 64 tokens long; then, for each token sequence in the training set, we ask the model to predict the next 64 tokens. We go through all of the original training dataset and produce an artificial dataset of the same size. Because we go through all of the original dataset and predict all of the blocks, if the model had 0 error, it would produce the original wikitext2 dataset. Training for each generation starts with generation from the original training data. Each experiment is run five times and the results are shown as five separate runs with different randomness seeds. The original model fine-tuned with real wikitext2 data obtains 34 mean perplexity, from the zero-shot baseline of 115, that is, it successfully learns the task. Finally, to be as realistic as possible, we use the best-performing model on the original task, evaluated using the original wikitext2 validation set, as the base model for the subsequent generations, meaning that—in practice—observed model collapse can be even more pronounced. Here we consider two different settings:

- Five epochs, no original training data. Here the model is trained for five epochs starting on the original dataset but with no original data retained for subsequent runs. The overall original task performance is presented in Fig. 1b. We find that training with generated data allows us to adapt to the underlying task, losing some performance, from 20 to 28 perplexity points.
- Ten epochs, 10% of original training data preserved. Here the model is trained for ten epochs on the original dataset and with every new generation of training, a random 10% of the original data points is sampled. The overall original task performance is presented in Fig. 1c. We find that preservation of the original data allows for better model fine-tuning and leads to only minor degradation of performance.

Both training regimes lead to degraded performance in our models, yet we do find that learning with generated data is possible and models can successfully learn (some of) the underlying task. In particular, from Fig. 1 and their 3D versions in the Supplementary Materials, we see that model collapse occurs, as the density of samples with low perplexity begins to accumulate over the generations. This in turn makes it likely that, over the generations, the sampled data will similarly collapse to a delta function.

It is important to note here that the observed behaviour is in line with the general intuition established in the section 'Theoretical intuition'. To be precise, in all experiments, generational learning is only performed on a finite (usually small) number of generations, whereas claims of the section 'Theoretical intuition' are mostly presented in the limit of generations going to infinity. However, as seen from experiments on VAEs and GMMs in the Supplementary Materials, convergence to delta functions and specific rates of such convergence are highly related to the specifics of the problem considered, and complete collapse may or may not occur, even after a small number of steps. This is further illustrated theoretically in the Supplementary Materials, in which potentially notable divergence from the original model can occur even after a few generations.

Figure 1b,c on the left shows histograms of individual data-point perplexities generated by the models of different generations as is evaluated by the first model developed with real wikitext2 training data. Here, over the generations, models tend to produce more sequences that the original model would produce with the higher likelihood. The observed effect is similar to that described for VAEs and GMMs in the Supplementary Materials, in which—over the generations—models started to produce samples that would be produced with higher probabilities by the original model. At the same time, we discover that generated data have much longer tails, suggesting that some of the data would never be produced by the original model—these are the errors that accumulate because of the learning with generational data.

Example 1. Example of text outputs of an OPT-125m model affected by model collapse—models degrade over generations, for which each new generation is trained on data produced by the previous generation.

- **Input:** some started before 1360 — was typically accomplished by a master mason and a small team of itinerant masons, supplemented by local parish labourers, according to Poyntz Wright. But other authors reject this model, suggesting instead that leading architects designed the parish church towers based on early examples of Perpendicular.
- **Outputs:**
- **Gen 0:** Revival architecture such as St. John's Cathedral in London. The earliest surviving example of Perpendicular Revival architecture is found in the 18th @-@ century Church of Our Lady of Guernsey, which dates from the late 19th century. There are two types of perpendicular churches : those.
- **Gen 1:** architecture such as St. Peter's Basilica in Rome or St. Peter's Basilica in Buenos Aires. There is no evidence that any of these buildings were built during the reign of Pope Innocent III, but it is possible that they may have been built during the reign of his successor, Pope Innocent.
- **Gen 5:** ism, which had been translated into more than 100 languages including English, French, German, Italian, Spanish, Portuguese, Dutch, Swedish, Norwegian, Polish, Hungarian, Slovak, Lithuanian, Estonian, Finnish, Romanian, Bulgarian, Turkish, Croatian, Serbian, Ukrainian, Russian, Kazakh, Kyrgyz.
- **Gen 9:** architecture. In addition to being home to some of the world's largest populations of black @-@ tailed jackrabbits, white @-@ tailed jackrabbits, blue @-@ tailed jackrabbits, red @-@ tailed jackrabbits, yellow @-.

Ablation: Repetitions

We find that data generated by language models in our experiments end up containing a large number of repeating phrases, as in Example 1. The repeating problem has been observed in nearly all text-generation models[17,18] and, to rule this out as the cause of model collapse, we further provide numerical experiments when models are explicitly encouraged to produce non-repeating sequences with a repeating penalty of 2.0. We find that this causes the models to produce lower score continuations to avoid using repeats, which—as a result—causes the consequent models to perform even worse. Model perplexities shift across the generations towards more probable token sequences, as measured using the model trained on the original real data distribution. Further illustrations are provided in the Supplementary Materials. In particular, enforcing this for the LLM experiments causes the perplexity to double compared with the original. Models remain as susceptible to model collapse, if not more.

The described process demonstrates that fine-tuning of language models does not curb the effects of model collapse and models that are being fine-tuned are also vulnerable. We find that, over the generations, models tend to produce more probable sequences from the original data and start introducing their own improbable sequences, that is, errors.

Discussion

We now discuss the implications of model collapse on the underlying learning dynamics of LLMs. Long-term poisoning attacks on language models are not new. For example, we saw the creation of click, content and troll farms, a form of human 'language models', whose job is to misguide social networks and search algorithms. The negative effect that these poisoning attacks had on search results led to changes in search algorithms. For example, Google downgraded farmed articles[19], putting more emphasis on content produced by trustworthy sources, such as education domains, whereas DuckDuckGo removed them altogether[20]. What is different with the arrival of LLMs is the scale at which such poisoning can happen once it is automated. Preserving the ability of

Content courtesy of Springer Nature, terms of use apply. Rights reserved

LLMs to model low-probability events is essential to the fairness of their predictions: such events are often relevant to marginalized groups. Low-probability events are also vital to understand complex systems[21].

Our evaluation suggests a 'first mover advantage' when it comes to training models such as LLMs. In our work, we demonstrate that training on samples from another generative model can induce a distribution shift, which—over time—causes model collapse. This in turn causes the model to mis-perceive the underlying learning task. To sustain learning over a long period of time, we need to make sure that access to the original data source is preserved and that further data not generated by LLMs remain available over time. The need to distinguish data generated by LLMs from other data raises questions about the provenance of content that is crawled from the Internet: it is unclear how content generated by LLMs can be tracked at scale. One option is community-wide coordination to ensure that different parties involved in LLM creation and deployment share the information needed to resolve questions of provenance. Otherwise, it may become increasingly difficult to train newer versions of LLMs without access to data that were crawled from the Internet before the mass adoption of the technology or direct access to data generated by humans at scale.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-024-07566-y.

1. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI blog* **1**, 9 (2019).
2. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
3. OpenAI. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf (2023).
4. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. in *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (eds Burstein, J., Doran, C. & Solorio, T.) 4171–4186 (Association for Computational Linguistics, 2019).
5. Liu, Y. et al. RoBERTa: a Robustly Optimized BERT Pretraining Approach. Preprint at https://arxiv.org/abs/1907.11692 (2019).
6. Zhang, S. et al. Opt: open pre-trained transformer language models. Preprint at https://arxiv.org/abs/2205.01068 (2022).
7. Aljundi, R., Kelchtermans, K. & Tuytelaars, T. Task-free continual learning. in: *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 11254–11263 (IEEE, 2019).
8. Carlini, N. & Terzis, A. in *Proc. Tenth International Conference on Learning Representations* (ICLR, 2022).
9. Carlini, N. et al. in *Proc. 2024 IEEE Symposium on Security and Privacy (SP)* 179 (IEEE, 2024).
10. Mousavi-Hosseini, A., Park, S., Girotti, M., Mitliagkas, I. & Erdogdu, M. A. in *Proc. Eleventh International Conference on Learning Representations* (ICLR, 2023).
11. Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S. & Srebro, N. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.* **19**, 1–57 (2018).
12. Gu, Y., Dong, L., Wei, F. & Huang, M. in *Proc. Twelfth International Conference on Learning Representations* (ICLR, 2024).
13. Shumailov, I. & Shumaylov, Z. Public code for Model Collapse (0.1). *Zenodo* https://doi.org/10.5281/zenodo.10866595 (2024).
14. Bommasani, R. et al. On the opportunities and risks of foundation models. Preprint at https://arxiv.org/abs/2108.07258 (2022).
15. Strubell, E., Ganesh, A. & McCallum, A. in *Proc. 57th Annual Meeting of the Association for Computational Linguistics* (eds Korhonen, A., Traum, D. & Màrquez, L.) 3645–3650 (Association for Computational Linguistics, 2019).
16. Merity, S., Xiong, C., Bradbury, J. & Socher, R. in *Proc. 5th International Conference on Learning Representations* (ICLR, 2017).
17. Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C. & Socher, R. CTRL: a conditional transformer language model for controllable generation. Preprint at https://arxiv.org/abs/1909.05858 (2019).
18. Shumailov, I. et al. in *Proc. 2021 IEEE European Symposium on Security and Privacy (EuroS&P)* 212–231 (IEEE, 2021).
19. Google. Finding more high-quality sites in search. *Google* https://googleblog.blogspot.com/2011/02/finding-more-high-quality-sites-in.html (2011).
20. Mims, C. The search engine backlash against 'content mills'. *MIT Technology Review* https://www.technologyreview.com/2010/07/26/26327/the-search-engine-backlash-against-content-mills/ (2010).
21. Taleb, N. N. Black swans and the domains of statistics. *Am. Stat.* **61**, 198–200 (2007).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2024, corrected publication 2025

Content courtesy of Springer Nature, terms of use apply. Rights reserved

Data availability

Data generation code for GMM experiments is available in ref. 13. Data used for VAE experiments are available in ref. 22. Data used for LLM experiments are available in ref. 16.

Code availability

Code for all experiments is publically available in ref. 13.

22. LeCun, Y., Cortes, C. & Burges, C. J. C. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (1998).

Acknowledgements This paper is dedicated to the memory of Professor Ross J. Anderson, our colleague and friend, who contributed much to this and other works we have produced over the years. We thank A. Thudi, D. Glukhov, P. Zaika, and D. Barak for useful discussions and feedback.

Author contributions I.S. and Z.S. proposed and developed the idea, led the research and mathematical modelling and developed the GMM and VAE experiments. I.S. and Y.Z. developed the language-model experiments. N.P., Y.G. and R.A. supervised and guided the project. All authors contributed to writing of the manuscript. Y.G. is supported by a Turing AI Fellowship financed by the UK government's Office for Artificial Intelligence, through UK Research and Innovation (grant reference EP/V030302/1) and delivered by the Alan Turing Institute.

Competing interests The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-024-07566-y.
Correspondence and requests for materials should be addressed to Ilia Shumailov, Zakhar Shumaylov, or Yarin Gal.
Peer review information *Nature* thanks the anonymous reviewers for their contribution to the peer review of this work.
Reprints and permissions information is available at http://www.nature.com/reprints.

Content courtesy of Springer Nature, terms of use apply. Rights reserved

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com